High Precision Testbed to Evaluate Ethernet Performance for In-Car Networks

Revsbech, Kasper; Madsen, Tatiana Kozlova; Schiøler, Henrik

Published in:
ITS Telecommunications (ITST), 2012 12th International Conference on

DOI (link to publication from Publisher):
10.1109/ITST.2012.6425239

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

High Precision Testbed to Evaluate Ethernet Performance for In-Car Networks

Kasper Revsbech, Tatiana K. Madsen, Henrik Schiøler
Electronic Systems
Aalborg University, Denmark
Email: {kar, tatiana, henrik}@es.aau.dk

***Abstract*—Validating safety-critical real-time systems such as in-car networks often involves a model-based performance analysis of the network. An important issue performing such analysis is to provide precise model parameters, matching the actual equipment. One way to obtain such parameters is to derive them by measurements of the equipment. In this work we describe the design of a testbed enabling active measurements on up to $1\ [Gb/Sec]$ Copper based Ethernet Switches. By use of the testbed it self, we conduct a series of tests where the precision of the testbed is estimated. We find a maximum error of $\pm 55\ [ns]$ measuring frame traversal time, and quantify the constant error imposed by the testbed it self.**

***Index Terms*—Safety-critical networks, Ethernet Modeling, In-car Networks, Delay Measurement, Performance Evaluation.**

I. Introduction

As in-car networks facilite a number of different traffic types, e.g: driver comfort messages, multimedia traffic and safety related messages, they must both provide a high bandwidth and hard bounds on traversal delays. Typically an in-car network consists of a variety of different bus technologies such as: CAN, FlexRay, MOST. Each applied because of its capabilities handling a specific type of traffic. However, the growing number of flows and the necessity of delivering the same message in various networks, has led to complex designs with a number of gateways bridging between the different technologies. As a consequence the Industry are investigating the feasibility of an Ethernet based networking instead [1], [2].

Since in-car networks are safety-critical they must be validated/verified to deliver a certain service to the critical flows. As Ethernet does not natively provide such guaranties validation/verification methods must be developed. In this work we have chosen to focus mainly on the validation and test of end-to-end delays.

A number of well established methods to evaluate the delay bounds in real-time networks already exists. An example is Real-Time Calculus (RTC), which is a mathematical approach of constructing network models based on max-plus algebra [3]. RTC models can be used computing delay and backlog bounds, providing valuable tools in the design and validation phase of safety-critical networks. However, a remaining challenge of using such methods is to obtain precise input parameters to the models as the parameters characterizing behavior of network equipment are rarely available. Commercial vendors of Ethernet switches in general do not specify the worst case bounds of their equipment and in many cases it is not possible to acquire the precise specifications of the internal design. Hence, switches are normally modelled in an abstract manner using well know scheduling models, e.g. FIFO [4]. We believe that by high precision testing, with various traffic scenarios the internal mechanics can be estimated and one can derive more precise model parameters.

In both [5] and [6] a similar idea is presented, where high precision measurements are used to estimate the model parameters of routers. In both cases high precision time-stamps are obtained by tapping the traffic ingress and egress on a router using DAG cards from Endace [7]. These time-stamps are then used to estimate the parameters for the network calculus server models: Guaranteed Rate (GR) and Packet Scale Rate Guarantee (PSRG). Similarly [8] presents a series of router measurements that are obtained using the same technology, to enable performance estimation. In this work we present a similar test facility, based on the same hardware principles, where a dedicated high precision networking card is used to timestamp the traffic. However, the presented testbed is also capable of generating traffic and timestamp it upon transmission to avoid the necessary classification as described in [8]. This follows the ideas in [9] where a NetFPGA card [10] is used as frame generator and receiver. Furthermore, the presented solution is developed specifically to allow high-precision measurements below the μsec range. In [5], [6] the measured processing times are within the ms range and in [8] a maximum error on the measurements is $2.2\mu sec.$ As we have designed the testbed by using only one adapter as transmitter and receiver we can achieve a higher precision as synchronization among cards can be avoided. Furthermore, we contribute with a set of experiments where the testbed it self is used to quantify its precision and the constant delay overhead of the measurements. Obviously such validation and quantification of the testbed is necessary when conducting measurements on safety-critical networks and when seeking to achieve sub μsec precision. Moreover, we believe that the experiments are generally applicable to similar testbeds/test setups e.g. those used in [5], [6] and could be applied if these should be used validating safety-critical networks.

In the remainder of this paper we first present the design of the testbed. This is followed by Section III that contains a series of test methods and test results to estimate the precision of the testbed. Furthermore in Sec III measurements on a real-life switch are shown. Finally Section V provides concluding remarks.

II. Testbed Design

The automotive industry currently focuses on the copper based 100BASE-TX technology, possibly in future we might see the usage of 1000BASE-T as well. Hence we have chosen to use the *NT4E* 4-port adapter from Napatech [11], which is equipped with four 1 $[Gb/sec]$, full-duplex, Small form-factor Pluggable (SFP) slots. In the current configuration we use 100BASE-TX/1000BASE-T transceivers. The *NT4E* adapter is a specially designed FPGA based networking interface card similar to the ones from Endace [7] and the NetFPGA project [10].

Since we would like to quantify switch traversal delays, the time precision of the testbed needs to be sufficient to quantify such delays. Observing a regular switch providing line speed service (1 $[Gb/sec]$), the minimum traversal time is at least the store-and-forward delay in the input buffer. For a minimum standard Ethernet frame of 64 [Bytes][1] this traversal time is: $(64 * 8)[bit]/1e9\ [bit/sec] \approx 0.5\ [\mu sec]$. As described in [12] using a standard Ethernet adapter in commodity hardware is not sufficient to meet the required timing precision. Hence the FGPA based special purpose cards is a feasible choice for such applications.

The chosen *NT4E* adapter from Napatech is equipped with high precision timers, with a resolution of 10 $[ns]$ [13]. Furthermore the adapter is capable of both receiving and transmitting frames, with the ability to time-stamp incoming and outgoing frames. It has been chosen to use the software appliance FT by Unispeed [14]. FT is an already existing software framework that facilitates the communication with the adapter and enable us to conduct line speed packet logging and generation.

Traversal delays are found by sending frames containing a time-stamp (TS_{send}) and by recording the time-stamps when receiving frames TS_{recv}. In that way the delay can be found as: $TS_{\Delta} = TS_{recv} - TS_{send}$. The TS_{send} time-stamp, of 64 $[bit]$, is inserted at a configurable offset by the adapter upon transmission. Likewise the adapter time-stamp receiving frames with a 64 $[bit]$ time-stamp (TS_{recv}). It is possible to configure if the adapter should time-stamp on first or last octet of received frames.

As described above the adapter is equipped with 4 full duplex ports, which means that it can log and transmit on all four ports simultaneously. This allows that the testbed both transmit and receive the flow(s) measured. Furthermore, due to the high-precision timers cross traffic can be generated in a very accurate way.

As the sole purpose of the testbed is to generate frames, of various sizes and inter-arrival times, and measure the traversal times, only time-stamps are logged to disk. It can be configured which to log, where the posible time-stamps to chose from are: TS_{send}, TS_{recv} and TS_{Δ}. Due to the nature of the tests a high amount of frames can arrive every second. As there is an one-to-one relation between arriving frames and log entries, small frames create more log entries. As an example logging 200 $[Bytes]$ frames at a rate of 1 $[Gb/sec]$ would generate $4e7$ log entries per second. Hence for small frames it can necessary to only log one parameter or/and to under-sample.

[1] By specification in IEEE 802.3 the minimum ethernet frame size is 64 [Bytes]

The traffic send from the adapter is generated according to the parameters specified in an configuration file, these parameters are:

- **IFG** - The inter frame gap between frames in nano seconds, which can be used to control the bandwidth of the flow.
- **Exponential Distribution** - If set to true, the **IFG** paramater is used as the mean in an exponential distributed inter arrival time.
- **Rate limit** - If applied the traffic is shaped to this band width.
- **Frame Size** - The frame size (Including Ethernet header) in Bytes.
- **Source and Destination MAC address** - To be able to control the receiver of a flow MAC Src. and Dst. can be configured.
- **Source and Destination UDP port** - All generated frames contain an IP and an UDP header. Where both the IPs and UDP ports can be configured.
- **Number of frames** - Specifies how many frames should send for this particular instance.

As seen from the parameters the testbed can send with a fixed Inter Frame Gap (IFG) or use an exponential distributed IFG. Furthermore, a rate limit can be applied. This allows transmission of both constant bit-rate and varying bit-rate traffic with bursts. This is in particular interesting when configuring the cross flows in a measurement scenario as it enables various realizations of the same traffic envelope.

III. Precision Estimation

As it is inevitable for the testbed to add some noise to the measurements in terms of timing variation, drift, etc, we are interested in quantifying this noise effect.

Common for all tests is, that we have used the testbed itself to estimate the precision, which can only be done measuring on a known deterministic system. In this case the deterministic systems are plain Cat. 5e twisted pair copper cables of various length. Of course the cable behavior is not strictly deterministic, as bit errors can occur and propagation time might vary as an effect of the temperature. However, the effects on the propagation time are considered insignificant, and as described in Sec. II it is possible to monitor the amount of send and received frames on each port, hence packet loss due to bit errors will be observed.

A. Max Deviation Estimation

Due to the nature of the HW timer(s) used when frames are time-stamped by the Ethernet adapter, variations in the frame traversal time estimates will occur. As described in Sec.

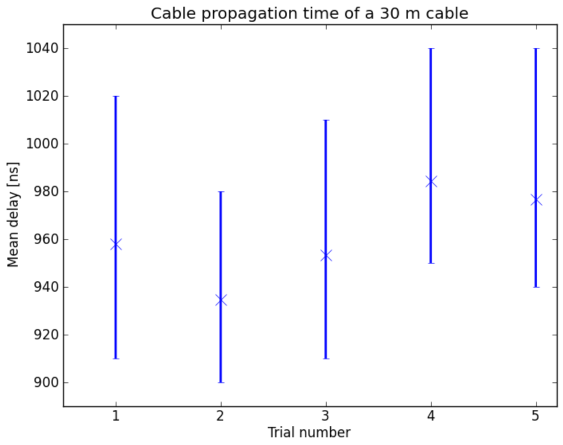


Fig. 1. Cable propagation times measured for a 30m cat 5e copper cable with independent experiment runs, each containing 1e6 samples.

II the resolution of the time-stamps provided by the adapter is 10 $[ns]$, however we still need to quantify the precision and variation of the time-stamps. Fig. 1 depicts mean delay of frames traversing a 30 [m] cat 5e cable. We show 5 repetitions each based on measurement of 1e6 frames with a size of 600 [Bytes], send at 1 [Gb/sec]. The error bars indicate the minimum and the maximum observed frame traversal time for each repetition.

As evident from Fig. 1, there are variations in the measurements, which could be quantized by a variance analysis assuming some underlying probabilistic distribution. However, as seen in Fig. 2 which are the histograms of the traversal times measures for the 5 repetitions, there is not a clear interpretation of the distribution. This indicates that the problem of modeling the distribution of the traversal times is not simple and might require more testing. However, one should note that in practice it is typically not the whole distribution we are interested in, but just its extremes. However, estimating the worst case sample error is a more relevant problem while designing safety-critical systems where worst case performance is the key metric.

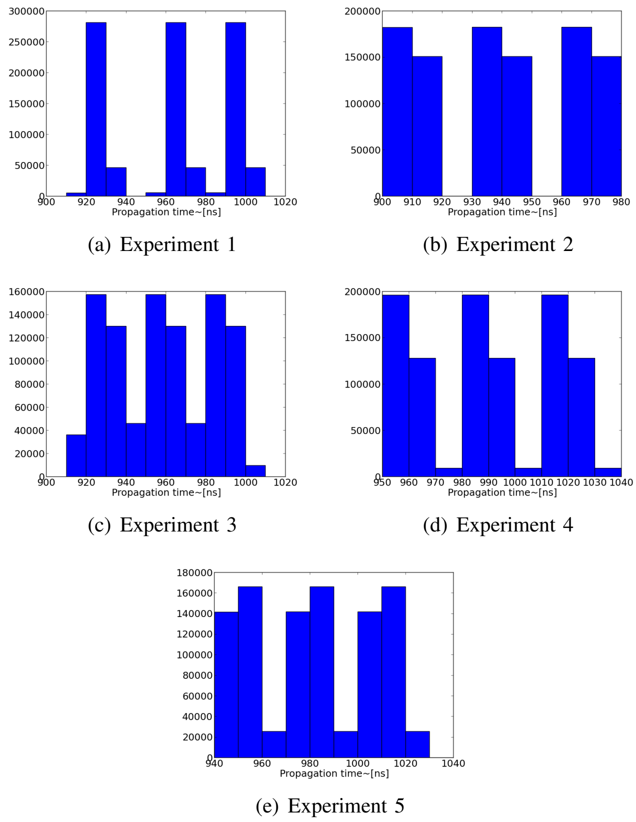
To quantify the maximum measurement error/noise, an experiment has been conducted where a large quantity of frames ($6e8$), with a size of 1400 $[Bytes]$ are send at 1 $[Gb/sec]$. thus the experiment has been running for approximately 112 minutes. Note that the frame size is not particular important. Again the frame traversal time of a 30 $[m]$ cable is measured, and the maximum sample error S_e is found as:

$$S_e = S_{max}\ [ns] - S_{min}\ [ns]$$

Where S_{max} and S_{min} are the maximum sample and minimum sample.
For this experiment we find that:

$$S_e = 1030 - 920 = 110\ [ns]$$

Hence we can conclude that for x_i from X the bounds becomes: $x_i \pm 55$ $[ns]$



(a) Experiment 1
(b) Experiment 2
(c) Experiment 3
(d) Experiment 4
(e) Experiment 5

Fig. 2. Histograms of the propagation times for the 5 experiments depicted in Fig. 1

B. Delay Overhead Estimation

In addition to the variations of the frame traversal times estimates found above, the testbed has a constant delay overhead, due to the TX-timestamp offset described in Sec. II and other constant factors. In this experiment we find this constant delay overhead by measuring the frame traversal time of cables with various lengths in order to estimate the frame traversal time as a function of cable length.

The experiment has been constructed in the following way: All cables used are Cat 5e cables which are connected directly between two ports in the adapter, $1e6$ frames of 600 $[Bytes]$ are then send from one port to another at 1 $[Gb/sec]$ and the traversal time of each frame is logged. The adapter is configured to place the TX time-stamp from Byte 40 and forward in the frame. Furthermore, the adapter is configured to obtain the RX time-stamp after receiving the first octet. Five independent trials have been conducted for each cable length, where the cable lengths used in the experiment are: [5 m, 10 m, 20 m, 30 m, 40 m, 50 m, 60 m, 100 m]. In Fig 3 the measured cable traversal times are shown, for each cable length the mean frame traversal time is estimated as the mean of the sample mean for the five repetitions. The error bars show the minimum and maximum observed traversal time of all 5 repetitions.

To estimate the traversal time as a function of the cable

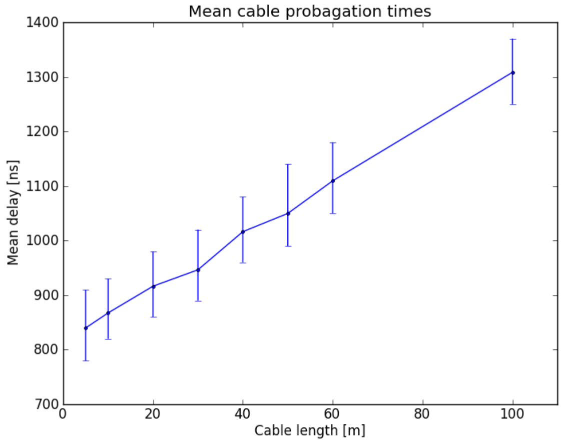


Fig. 3. Propagation time measured on Cat 5e copper cables of various lengths

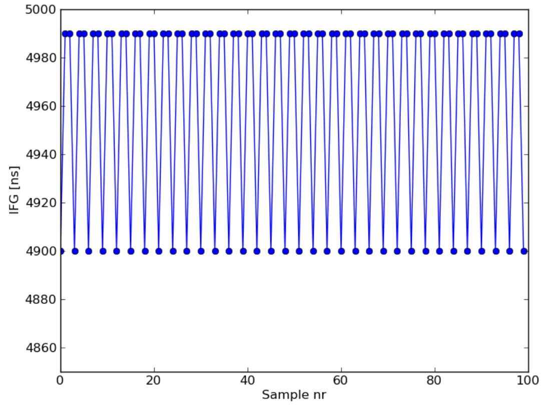


Fig. 4. The first 100 samples of the frame generation test at line speed (1 $[Gb/sec]$)

length we have used a first order linear regression using the mean estimate for each cable length. We estimate $a = 4.9\ ns$ and $b = 813.3\ ns$, hence we can make the following relationship between cable length and traversal time:

$$f(x) = 4.9\ x + 813.3$$

Where x is the cable length in $[m]$, and $f(x)$ is the measured cable traversal time in $[ns]$. From the specifications of a Cat 5e cable [15] we see that the propagation delay is 5.3 $[ns]$ per meter at $20°C$, hence we conclude that our test results are aligned with the specifications. With respect to delay overhead we estimate it to be 813.3 $[ns]$ as this is the delay at "zero" cable length. One should note that the overhead is constant as long at the TX time-stamp offset remains at the same place and the RX time-stamp is conducted on first octet. Hence the delay overhead can be subtracted from the measured delays.

C. *Frame Generation Test*

As described in Sec. II, the frame generation tool can be configured to send a certain amount of frames with a certain IFG. In this experiment we test if the testbed can actually send the frames at exactly the specified IFG. We test this by logging the TX time-stamps generated by the testbed it self. We have constructed two scenarios where we in both cases send 1e6 frames: One where we send at line speed (1 $[Gb/sec]$), and one where the IFG is 24000 $[ns]$ with a frame size of 600 $[B]$ which gives a bandwidth of 200 $[Mb/sec]$. In Fig 4 we show a zoom of the first 100 samples for the line speed scenario, as seen from the figure the TX time has a small variation. For the line speed test we observed a maximum deviation of 90 $[ns]$, and a mean bandwidth of 967 $[Mbit/sec]$. In the experiment with an IFG of 24000 $[ns]$ we found a max deviation of 90 $[ns]$ and a mean bandwidth of 200 $[Mbit/sec]$.

IV. Examples on Cisco switch

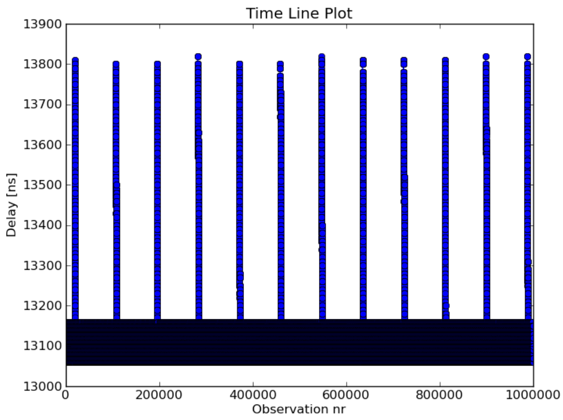
Having quantized the precision of testbed it self, we would like to show how it can be used to measure on an actual Ethernet switch. In this experiment we have chosen a 5 port *Cisco SD 2005* 10/100/1000 switch. We send one flow trough the switch using two ports on the adapter. It has been chosen to send 1e6 frames of 1400 $[Bytes]$ at 1 $[Gb/sec]$. Since the switch will broadcast traffic if the receiving MAC address has not "been seen" we send 10 frames from the receiving port before we start the measured flow.

In Fig. 5(a) the obtained delays are plotted. As seen from the figure there are a number of spikes in the measured delays. Furthermore, studying the statistics obtained after test we see that 1e6 frames have been sent. However, we could see that we received 1000033 frames. As described above we know that the first 10 frames are broadcast frames. By packet sniffing of the scenario we have found that the rest of the frames are spanning tree protocol packets generated by the switch it self. This explains the spikes because frames are queued when such a frame is broadcasted on the switch. In Fig. 5(b) we see a zoom of one of the spikes, and as seen from the figure the delay rises suddenly as an effect of the spanning tree protocol frame. However the observed delay is slowly reducing again and after $\approx$ 1500 frames it is back to normal. One would expect this delay to remain the same as we keep filling the queue with frames at 1 $[Gb/sec]$. However, as described in Sec. III-C the reduction of the delay is due to the fact that the frame generator is a bit slower than 1 $[Gb/sec]$ hence the queue is decreased over time.

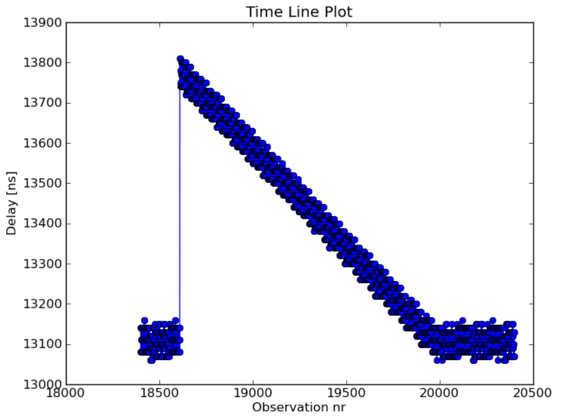
From Fig. 5(a) and 5(b) we also see the that the delay from store and forward varies from sample to sample. In Fig. 5(b) we show the first 1e3 samples, and as seen from the figure the difference is the same as quantized in Sec. III-A hence it seems to be due to the variations in testbed it self, and not due to the nature of the switch.

V. Conclusion and Outlook

The design of a high-precision testbed, targeting validation of Ethernet networks used in safety-critical systems, has been presented. The testbed is capable of both receiving and transmitting frames at up to 1 $[GB/sec]$ on four ports. It is primarily designed to enable a series of experiments



(a) One flow



(b) Zoom on the first delay spike in Fig. 5(a)

Fig. 5. Switch test with one flow

on Ethernet switches to be able to obtain precise model parameters, which are especially important in the design and validation of in-car networks. However, as the testbed can also be configured to log RX and TX time-stamps and has a configurable traffic generation, the application domain is not limited to safety-critical networks.

We have presented a series of tests where the precision of the testbed has been quantized. This has high importance as it is necessary to know it the observed affects are an effect of the measured system, or the testbed it self. We believe that the presented results are interesting to the research community as an indicator of what can be achieved using an FPGA based adapter. Experiments on precision estimation described here can be applied to any similar testbeds, based on similar hardware/software or e.g. the open netFPGA platform or Endace based systems. From the obtained measurements presented in Sec. III-A we see that the maximum deviation of one traversal time due to the nature of the testbed is ± 55 $[ns]$, which we believe is sufficient to study effects on a Ethernet switch.

ACKNOWLEDGMENT

The authors would like to thanks Unispeed A/S for the collaboration in the design and implementation of the testbed. Furthermore we would like to thank Kim Højgaard-Hansen for valuable input to the design and implementation of the testbed and the inputs for this manuscript.

REFERENCES

[1] R. Steffen, R. Bogenberger, J. Hillebrand, W. Hintermaier, A. Winckler, and M. Rahmani, “Design and realization of an ip-based in-car network architecture,” in *The First Annual International Symposium on Vehicular Computing System*, (Dublin, Ireland), July 22-24, 2008.

[2] J. Rox, R. Ernst, and P. Giusto, “Using timing analysis for the design of future switched based Ethernet automotive networks,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*, pp. 57–62, 2012.

[3] E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse, “System architecture evaluation using modular performance analysis: a case study,” *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 8, Oct 2006.

[4] K. Revsbech, H. Schiøler, T. K. Madsen, and J. J. Nielsen, “Worst-Case Traversal Time Modelling of Ethernet Based In-Car Networks Using Real Time Calculus.,” *NEW2AN*, pp. 219–230, 2011.

[5] A. Undheim, P. Emstad, and Y. Jiang, “Network Calculus Approach to Router Modeling with External Measurements,” *q2s.ntnu.no*, pp. 1–5, Mar. 2007.

[6] M. Bredel, Z. Bozakov, and Y. Jiang, “Analyzing router performance using network calculus with external measurements,” in *2010 IEEE 18th International Workshop on Quality of Service (IWQoS)*, pp. 1–9, IEEE, 2010.

[7] “Endace homepage.” www.endace.com.

[8] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot, “Bridging router performance and queuing theory,” in *SIGMETRICS ’04/Performance ’04: Proceedings of the joint international conference on Measurement and modeling of computer systems*, ACM Request Permissions, June 2004.

[9] G. Covington, G. Gibb, J. Lockwood, and N. Mckeown, “A Packet Generator on the NetFPGA Platform,” *Field Programmable Custom Computing Machines, 2009. FCCM ’09. 17th IEEE Symposium on*, pp. 235–238, 2009.

[10] “The netfpga project homepage.” netfpga.org.

[11] “Napatech homepage.” http://www.napatech.com/.

[12] J. Cleary, I. Graham, T. McGregor, M. Pearson, L. Ziedins, J. Curtis, S. Donnelly, J. Martens, and S. Martin, “High precision traffic measurement,” *IEEE Communications Magazine*, vol. 40, pp. 167–173, Mar. 2002.

[13] “Nt4e data sheet.” http://www.napatech.com/uploads/c_file50_file_9609.pdf.

[14] “Unispeed homepage.” http://unispeed.com/.

[15] “Cat 5e utp cable data sheet.” http://communications.draka.com/sites/eu/Datasheets/SuperCat5_24_U_UTP_Install.pdf.